

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-153080

(43)Date of publication of application : 11.06.1996

(51)Int.Cl. G06F 15/16
G06F 15/163

(21)Application number : 06-296416

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 30.11.1994

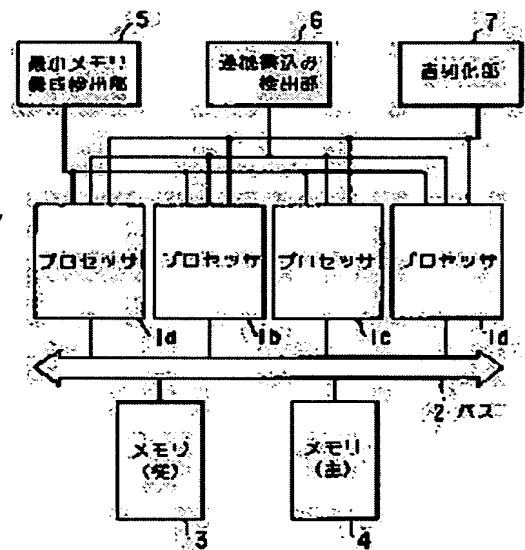
(72)Inventor : SEKIDO KAZUNORI

(54) MULTIPROCESSOR SYSTEM AND MEMORY ACCESS CONTROLLING METHOD

(57)Abstract:

PURPOSE: To reduce the drop of writing performance in the case where write to a memory by plural processors coincides with each other even in the case of minimum memory configuration.

CONSTITUTION: The system has processors 1a to 1d and the memories 3, 4, and has a fault resisting property against a memory fault realized by making the memories store the same data as classifying them as a sub memory 3 and a main memory 4, and executing the write of the data to the main memory 4 after the sub memory 3 and executing the read of the data from the main memory 4, and a continuous write detecting part 6 which detects the execution of continuous write that the write to the main memory 4 is executed in succession after the write of the data to the sub memory 3 was executed, and serializes the write so that the write to the sub memory 3 by another processor is executed after the write to the sub memory 3 by one processor is finished in the case where the execution of the continuous write to the common memory by two processors is detected by the continuous write detecting part 6.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-153080

(43) 公開日 平成8年(1996)6月11日

(51) Int.Cl.⁶

G 0 6 F 15/16
15/163

識別記号

4 7 0 M

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/ 16

3 2 0 M

審査請求 未請求 請求項の数7 O L (全 13 頁)

(21) 出願番号

特願平6-296416

(22) 出願日

平成6年(1994)11月30日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 関戸 一紀

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

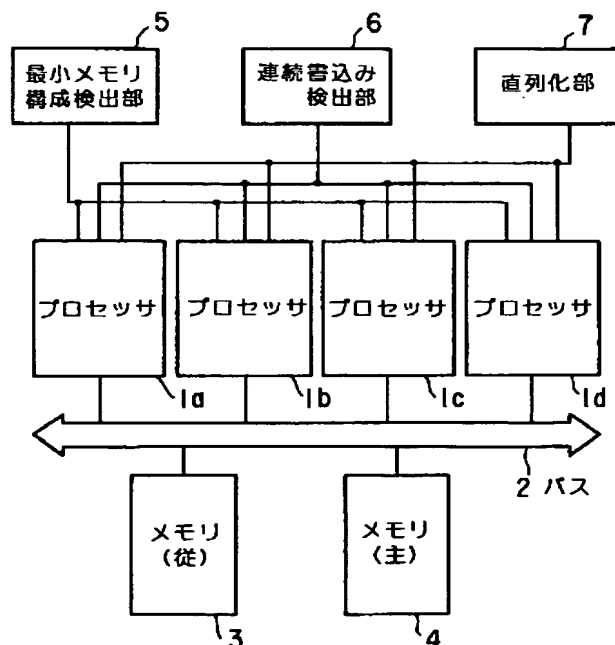
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 マルチプロセッサシステム及びメモリアクセス制御方法

(57) 【要約】

【目的】 最小メモリ構成であっても、複数のプロセッサによるメモリに対する書き込みが重なった場合の書き込みの性能低下を小さくすることを可能にする。

【構成】 プロセッサ1a～1dとメモリ3、4を有し、従のメモリ3と主のメモリ4に分けて同じデータを保持させ、の書き込みを従のメモリ3の次に主のメモリ4に対して行ない、読み込みを主のメモリ4から行なうことによりメモリ障害に対する耐故障性を有し、従のメモリ3にデータの書き込みを行なってから主のメモリ4に続けて書き込みを行なう連続書き込みを実行することを検出する連続書き込み検出部6と、連続書き込み検出部6によって共通するメモリに対する連続書き込みを2つのプロセッサが実行することが検出された場合に、一方のプロセッサによる従のメモリ3に対する書き込みが終わった後に他方のプロセッサによる従のメモリ3に対する書き込みを行なうように書き込みを直列化する。



1

【特許請求の範囲】

【請求項 1】 複数のプロセッサと複数のメモリを接続し、同時に 2 つ以上のメモリアクセスを処理できるバスを有するマルチプロセッサシステムであって、メモリを主と従に分けてそれぞれに同じデータを保持させるとともに、データの書き込みを従のメモリの次に主のメモリに対して行ない、データの読み込みを主のメモリから行なうことによりメモリ障害に対する耐故障性を有するマルチプロセッサシステムにおいて、
10 従のメモリにデータの書き込みを行なってから主のメモリに続けて書き込みを行なう連続書き込みを実行することを検出する連続書き込み検出手段と、
前記連続書き込み検出手段によって共通するメモリに対する連続書き込みを 2 つのプロセッサが実行することが検出された場合に、一方のプロセッサによる従のメモリに対する書き込みが終わった後に他方のプロセッサによる従のメモリに対する書き込みを行なうように書き込みを直列化する制御手段と、
を具備したことを特徴とするマルチプロセッサシステム。

【請求項 2】 前記複数のメモリは、従のグループの複数のメモリと、主のグループの複数のメモリとに分けられる場合に、従のグループのメモリが 1 つと従のグループのメモリが 1 つの 1 組しかない最小メモリであることを検出する最小メモリ構成検出手段をさらに具備し、最小メモリ構成検出手段によって最小メモリ構成である場合に、メモリに対する書き込みを直列化することを特徴とする請求項 1 記載のマルチプロセッサシステム。

【請求項 3】 1 つのプロセッサによる連続書き込みの開始時刻を保持する直列化手段を有し、
前記連続書き込み検出手段によって共通するメモリに対する連続書き込みを前記 1 つのプロセッサの他のプロセッサが実行することが検出された場合に、
前記直列化手段に保持されている開始時刻と従のメモリに対して既に実行している書き込みの時間から、残りの従のメモリ及び主のメモリに対する連続書き込み時間を求め、この連続書き込み時間と前記他のプロセッサが行なう書き込みの連続書き込み時間から、直列化した場合と直列化しない場合の書き込み時間を求め、直列化した場合の方が時間を要する場合に直列化しないで連続書き込みを行なうことを特徴とする請求項 1 記載のマルチプロセッサシステム。

【請求項 4】 前記連続書き込み検出手段は、1 つのプロセッサによる連続書き込みのメモリ衝突がない場合の連続書き込み時間を求める機能を有し、
前記連続書き込み検出手段によって求められた連続書き込み時間を保持する直列化手段を有し、
前記連続書き込み検出手段によって共通するメモリに対する連続書き込みを前記 1 つのプロセッサの他のプロセッサが実行することが検出された場合に、

2

前記直列化手段に保持されている連続書き込み時間と、これから前記他のプロセッサによって実行される書き込みの連続書き込み時間から、直列化した場合と直列化しない場合の書き込み時間の期待値を求め、この期待値に基づいて書き込みの直列化をするか否かを決定することを特徴とする請求項 1 記載のマルチプロセッサシステム。

【請求項 5】 複数のプロセッサと複数のメモリを接続し、同時に 2 つ以上のメモリアクセスを処理できるバスを有するマルチプロセッサシステムであって、メモリを主と従に分けてそれぞれに同じデータを保持させるとともに、データの書き込みを従のメモリの次に主のメモリに対して行ない、データの読み込みを主のメモリから行なうことによりメモリ障害に対する耐故障性を有するマルチプロセッサシステムにおいて、
10 従のメモリにデータの書き込みを行なってから主のメモリに続けて書き込みを行なう連続書き込みを、共通するメモリに対して 2 つのプロセッサが実行することが検出された場合に、一方のプロセッサによる従のメモリに対する書き込みが終わった後に他方のプロセッサによる従のメモリに対する書き込みを行なうように書き込みを直列化することを特徴とするメモリアクセス制御方法。

【請求項 6】 従のメモリにデータの書き込みを行なってから主のメモリに続けて書き込みを行なう連続書き込みを、共通するメモリに対して 2 つのプロセッサが実行することが検出された場合に、
1 つのプロセッサによる先に連続書き込みを開始した開始時刻と従のメモリに対して既に実行している書き込みの時間から、残りの従のメモリ及び主のメモリに対する連続書き込み時間を求め、この連続書き込み時間と後から連続書き込みを開始する他のプロセッサによる連続書き込み時間から、直列化した場合と直列化しない場合の書き込み時間を求め、直列化した場合の方が時間を要する場合に直列化しないで連続書き込みを行なうことを特徴とする請求項 5 に記載のメモリアクセス制御方法。

【請求項 7】 従のメモリにデータの書き込みを行なってから主のメモリに続けて書き込みを行なう連続書き込みを、共通するメモリに対して 2 つのプロセッサが実行することが検出された場合に、
40 先に連続書き込みを開始した 1 つのプロセッサによるメモリ衝突がない場合の連続書き込み時間と、後から連続書き込みを開始する他のプロセッサによる連続書き込み時間から、直列化した場合と直列化しない場合の書き込み時間の期待値を求め、この期待値に基づいて書き込みの直列化をするか否かを決定することを特徴とする請求項 5 に記載のメモリアクセス制御方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、メモリを主と従の 2 つのグループに分けて同じデータを保持させ、メモリ障害

3

に対する耐故障性を持たせたマルチプロセッサシステムに関する。

【0002】

【従来の技術】従来、マルチプロセッサシステム等の情報処理装置において、メモリの耐故障性を高める方法としては、同じデータを主と従の2グループのメモリに持たせる構成がある。この構成では、一方のメモリが故障し、その中のデータが全くなっても、他方のメモリに格納されたデータの内容が正しいことが保証されている。従って、故障したメモリを活線と交換し、故障したメモリに内容が保証された他方のメモリからデータをコピーすることにより、システムを止めることなく、メモリの故障に対処することが可能である。

【0003】この種の構成では、通常、主と従の2グループのメモリには同じデータが格納されている必要がある。そこで、主と従のメモリに同時に同じデータを書き込む方法が考えられるが、バースト転送でデータをメモリに書き込んでいる途中でプロセッサ等が故障すると、2つのメモリが両方とも途中状態になってしまうので、プロセッサ等が絶対に故障しないようハードウェアを3重化以上にしなければならず、コストが非常に高くなってしまふ。

【0004】もう1つの書き込み方法として、まず、従のメモリに書き込みを行ない正常に書き込みが完了してから、次に主のメモリへ同じデータを書き込み、データを読む時には必ず主のメモリから読むことにより、主と従の2グループのメモリにデータが書き込まれた事を保証する方法もある。この方法では、メモリ書き込み中にプロセッサが故障しても、主または従の何れか一方のメモリには、書き込まれる前の状態が書き込みが完了した状態のどちらかの状態が完全に残る。

【0005】そこで、メモリに書き込みを行なう時には、他プロセッサが処理を引継げるだけの情報も合わせてメモリに書き込んでおけば、完全に残っている内容が保証されたメモリの情報を引き継いで他のプロセッサが処理を続けられる。よって、プロセッサが連続書き込みの途中、すなわち従来のメモリに対する書き込みを行なった後、主のメモリに書き込みを行なう途中で故障しても、システムを回復させることができる。故に、ハードウェアコストを下げたシステムでは、従のメモリに書き込みを行なってから主のメモリに書き込みを行なう、前述した後者のメモリアクセス制御方法がとられている。

【0006】

【発明が解決しようとする課題】このような従来のメモリアクセス制御方法では、必ず従のメモリに書いてから主のメモリに書き込みが行なわれる。ここで、図11に示すような、2つのプロセッサA、Bが設けられ、メモリの主と従の2グループにおいて、それぞれにメモリが1つずつしかない最小メモリ構成の場合を考える。

【0007】この場合、一方のプロセッサA（あるいは

4

プロセッサB）によって、単独で連続書き込みを実行する場合、図12(a)に示すように、従のメモリへの書き込み、主のメモリへの書き込みが、連続して最短の時間で実行できる。

【0008】ところが、2つのプロセッサA、Bによるメモリへの書き込みが同時実行（開始のタイミングがほぼ一致したものとする）される状況となった場合には、図12(b)に示すようになる。

【0009】2つのプロセッサA、Bによる従のメモリに対する書き込みが重なった場合、従のメモリが1つしかない事から、従のメモリのボトルネックが発生する。すなわち、従のメモリは、プロセッサA、Bからの各書き込みを交互に1つずつしか処理できないことから、一方のプロセッサが書き込みを実行している間、同じ時間だけ他方のプロセッサは待ちが発生するため、1つのプロセッサによる従のメモリへのアクセス時間は、単独実行時の2倍要してしまう。

【0010】さらに、主のメモリに対する書き込みも同時に始まってしまい、主のメモリも1つしかない事から主のメモリのボトルネックも発生する。従のメモリの場合と同様に主のメモリは、プロセッサA、Bからの各書き込みを交互にしか処理できないので、主のメモリへのアクセスも単独実行時の2倍になってしまう。

【0011】以上のように、従来のメモリアクセス制御方法では、2つのプロセッサによる書き込みが1つのメモリについて重なった場合、従のメモリと主のメモリでアクセスが競合してしまい、書き込み時間を大きく増加させ、システム性能を低下させてしまうという問題があった。

【0012】本発明は前記のような事情を考慮してなされたもので、最小メモリ構成であっても、複数のプロセッサによるメモリに対する書き込みが重なった場合の書き込みの性能低下を小さくすることが可能なメモリアクセス方法を提供することを目的とする。

【0013】

【課題を解決するための手段】本発明は、複数のプロセッサと複数のメモリを接続し、同時に2つ以上のメモリアクセスを処理できるバスを有するマルチプロセッサシステムであって、メモリを主と従に分けてそれぞれに同じデータを保持させるとともに、データの書き込みを従のメモリの次に主のメモリに対して行ない、データの読み込みを主のメモリから行なうことによりメモリ障害に対する耐故障性を有するマルチプロセッサシステムにおいて、従のメモリにデータの書き込みを行なってから主のメモリに続けて書き込みを行なう連続書き込みを実行することを検出する連続書き込み検出手段と、前記連続書き込み検出手段によって共通するメモリに対する連続書き込みを2つのプロセッサが実行することが検出された場合に、一方のプロセッサによる従のメモリに対する書き込みが終わった後に他方のプロセッサによる従のメ

メモリに対する書き込みを行なうように書き込みを直列化する制御手段とを具備したことを特徴とする。

【0014】また前記複数のメモリは、従のグループの複数のメモリと、主のグループの複数のメモリとに分けられる場合に、従のグループのメモリが1つと従のグループのメモリが1つの1組しかない最小メモリであることを検出する最小メモリ構成検出手段をさらに具備し、最小メモリ構成検出手段によって最小メモリ構成である場合に、メモリに対する書き込みを直列化することを特徴とする。

【0015】また、1つのプロセッサによる連続書き込みの開始時刻を保持する直列化手段を有し、前記連続書き込み検出手段によって共通するメモリに対する連続書き込みを前記1つのプロセッサの他のプロセッサが実行することが検出された場合に、前記直列化手段に保持されている開始時刻と従のメモリに対して既に実行している書き込みの時間から、残りの従のメモリ及び主のメモリに対する連続書き込み時間を求め、この連続書き込み時間と前記他のプロセッサが行なう書き込みの連続書き込み時間から、直列化した場合と直列化しない場合の書き込み時間を求め、直列化した場合の方が時間を要する場合に直列化しないで連続書き込みを行なうことを特徴とする。

【0016】また前記連続書き込み検出手段は、1つのプロセッサによる連続書き込みのメモリ衝突がない場合の連続書き込み時間を求める機能を有し、前記連続書き込み検出手段によって求められた連続書き込み時間を保持する直列化手段を有し、前記連続書き込み検出手段によって共通するメモリに対する連続書き込みを前記1つのプロセッサの他のプロセッサが実行することが検出された場合に、前記直列化手段に保持されている連続書き込み時間と、これから前記他のプロセッサによって実行される書き込みの連続書き込み時間から、直列化した場合と直列化しない場合の書き込み時間の期待値を求め、この期待値に基づいて書き込みの直列化をするか否かを決定することを特徴とする。

【0017】

【作用】このような構成によれば、複数のプロセッサによる従のグループの1つのメモリへのアクセスを直列化することにより、2つの連続書き込みが重なって従のメモリ及び主のメモリでのアクセス衝突が可能な限り回避され、連続書き込みの書き込み時間の増加を抑えることができる。連続書き込みに要する時間が小さくなれば、その分、プロセッサを本来の処理の実行に使えるので、システム性能が向上される。また、状況に応じて直列化の有効性を判別して、その結果に基づいた制御を行なうので、最適化されたメモリアccessが可能となる。

【0018】

【実施例】以下、図面を参照して本発明の実施例を説明する。図1は本発明の第1実施例に係わるマルチプロセ

ッサシステムの構成を示すブロック図である。図1に示すように、第1実施例におけるマルチプロセッサシステムは、複数のプロセッサ1a～1dが、バス2を介してメモリ3、4と接続されて構成されている。

【0019】プロセッサ1a～1dは、メモリ3、4を共有している。各プロセッサ1a～1dは、メモリへの書き込みは、まず従のメモリに対して行ない、次に主のメモリに対して行なう。メモリからの読み出しは、主のメモリから行なう。

10 【0020】バス2は、プロセッサ1a～1d中の2つ以上のプロセッサからのメモリ3、4に対する同時のメモリアccessを処理できる。メモリ3は、プロセッサ1a～1dによって従のメモリとして用いられ、メモリの故障に対処するため、主のメモリ4と同じデータを保持する。各プロセッサ1a～1dによる連続書き込みの際には、メモリ3に対するメモリの書き込みが、メモリ4の前に行われる。

20 【0021】メモリ4は、プロセッサ1a～1dによって主のメモリとして用いられ、メモリの故障に対処するため、従のメモリ3と同じデータを保持する。各プロセッサ1a～1dによる連続書き込みの際には、従来のメモリ3に対する書き込みの後に行なわれる。プロセッサ1a～1dによるメモリの読み出しは、この主のメモリ4に対してだけ行われる。

【0022】さらに、本発明では最小メモリ構成検出部5、連続書き込み検出部6、及び直列化部7が設けられ、各プロセッサ1a～1dにおけるメモリアccess制御の際に用いられる。

30 【0023】最小メモリ構成検出部5は、バス2に接続されているメモリの構成を判別するもので、主のメモリ4と従のメモリ3が1組しかない最小のメモリ構成であることを通知するものである。

【0024】連続書き込み検出部6は、プロセッサ1a～1dのそれぞれによって実行される処理が、従のメモリ3と主のメモリ4に対して、連続してデータの書き込みを行なう連続書き込みであることを事前に知ることができる。連続書き込み検出部6は、プロセッサ1a～1dによるメモリアccessの際に、各プロセッサによって参照される。

40 【0025】直列化部7は、従のメモリ3に対する連続書き込みをプロセッサ1a～1dの1台しか行えないようにするために用いられるもので、従のメモリ3に対する書き込みを行なうプロセッサによって占有されることで、他のプロセッサが従のメモリ3に対する書き込みをできないことを示す排他変数を保持する。これにより、2つのプロセッサによる従のメモリ3に対する書き込みを直列化することができる。

50 【0026】図2は本発明のメモリアccess制御方法をマルチプロセッサシステム上で実現する機能構成を示すブロック図である。図2中の最小メモリ構成検出部5、

連続書き込み検出部6、及び直列化部7は、図1を用いて説明した各部と基本的には同様の機能を有するものとして同一符号を付している。

【0027】メモリアクセス制御部10は、各プロセッサ1a~1dにおいて実行されるもので、最小メモリ構成検出部5、連続書き込み検出部6、及び直列化部7を用いてメモリアクセスを制御する。

【0028】ハードウェア構成情報格納部12は、マルチプロセッサシステムにおけるシステム構成に関する情報を格納するもので、従のメモリと主のメモリに関する情報が含まれている。例えば、構成定義ファイルとしてメモリに格納される。構成定義ファイルは、メモリアクセス制御の中で最小メモリ構成検出部5によって判別され、システムの動作中にメモリ構成の変更と共に情報の内容を変更することによって、変更に応じたアクセス制御が可能である。マルチプロセッサシステムをフォールトトレラントシステムとした場合に好適である。また、システムの停止を容易に行なうことができれば、例えばブート時に最小メモリ構成検出部5にメモリ構成の情報を設定するようにしても良い。

【0029】連続書き込み検出部6は、例えば各プロセッサ1a~1dで実行されるメモリに書き込みを行なうプログラムによって、連続書き込みを行なうことを示す変数がセットされる。これにより、メモリアクセス制御部10に連続書き込みを通知することができる。なお、連続書き込みを行なうことを示す変数をセットするプログラムは、例えばコンパイル時にメモリに対する連続書き込みを行なうルーチンが存在した場合に、そのルーチンの先頭に変数をセットする手続きが付加されて実現される。

【0030】直列化部7は、各プロセッサ1a~1dでメモリアクセス制御部10を実行することによって、メモリアクセスしようとするプロセッサによって参照される。連続書き込み検出部6及び直列化部7は、メモリ上でそれぞれ、連続書き込みを行なうことを示す変数、排他変数を保持することによって実現されても良い。

【0031】次に、第1実施例の動作について説明する。第1実施例におけるプロセッサ1a~1dは、図3に示すフローチャートに従ってメモリアクセスを行なう。なお、プロセッサ1a~1dは、メモリにデータを書く場合には、まず従のメモリ3に対する書き込みが終わってから次に主のメモリ4に対して書き込みを行ない、メモリからデータを読み込む場合には、主のメモリ4からしか読まない。このように制御することで、プロセッサが参照するデータが常にメモリ3、4の両方に格納されていることを保証する。また、メモリに書く場合には、他のプロセッサが処理を引き継げるだけの情報を合わせてメモリへ連続書き込みで書くことにより、書き込み途中でプロセッサが故障しても完全な状態が残っている。メモリに格納されている情報を使って他のプロセ

ッサが処理を継続できる。

【0032】まず、メモリアクセスを行なうプロセッサ(メモリアクセス制御部10)は、連続書き込み検出部6を使って、メモリアクセスが連続書き込みか読み出しかを判断する(ステップA1)。

【0033】メモリアクセスが読み出しの場合には、プロセッサは、主のメモリ4から読み出しを行なう(ステップA2)。連続書き込みの場合には、最小メモリ構成検出部5により、主と従のメモリが1つずつしかなく、同時に連続書き込みを行なうとメモリでの転送ネックが発生する最小メモリ構成であるかを判別する(ステップA3)。

【0034】メモリが最小構成でなければ、プロセッサは(メモリアクセス制御部10による制御のもとに)、他のプロセッサが連続書き込みを行っている、いないにかかわらず、メモリに対する連続書き込み、すなわち従のメモリ3への書き込みを行ない(ステップA4)、次に主のメモリ4への書き込みを行なう(ステップA5)。

【0035】一方、メモリが最小構成の場合には、従のメモリ3への連続書き込みを直列化するため、プロセッサ(メモリアクセス制御部10)は、直列化部7に問い合わせ直列化部7が空いているか、つまり他のプロセッサが従のメモリ3への連続書き込みを行なっているかを調べる(ステップA6)。

【0036】空いていない場合には、プロセッサは、直列化部7を占有している他のプロセッサによる従のメモリ3への連続書き込みが終了し、直列化部7が解放されるまで待つ(ステップA7)。

【0037】一方、直列化部7が空いている場合には、プロセッサは、従来のメモリ3に対する書き込みを行なっている間は、他プロセッサによる従のメモリ3への連続書き込みを禁止させるため直列化部7を占有する(ステップA8)。ここで直列化部7が占有されると他のプロセッサは、従のメモリ3に対する書き込みを実行できず、直列化部7が解放されるまで待ち状態となる。

【0038】続いて、プロセッサは、従のメモリ3への連続書き込みを実際に開始する(ステップA9)。プロセッサは、従のメモリ3に対する書き込みが完了した後で直列化部7を解放し(ステップA10)、他のプロセッサが連続書き込みを開始できるようにする。その後、主のメモリ4への連続書き込みを行なう(ステップA11)。直列化部7が解放されると、空くまで待っていたプロセッサが、直列化部7を占有して、従のメモリ3に対する連続書き込みを開始する。

【0039】ここで、図1に示す最小メモリ構成のマルチプロセッサシステムで、プロセッサ1aによる連続書き込みAとプロセッサ1bによる連続書き込みBがほぼ同じに始まった場合について、簡単な例を用いて説明する。なお、説明の都合上、プロセッサ1aによる書き込みAの方が少しだけ早く開始されたものとし、プロセ

サ1 a, 1 bによる書き込みに要する時間はほぼ同じものとする。

【0040】プロセッサ1 a (あるいはプロセッサ1 b) によって、単独で連続書き込みを実行する場合、図4 (a) に示すように、従のメモリへの書き込み、主のメモリへの書き込みを、連続して最短の時間で実行できる。

【0041】これに対し、プロセッサ1 aとプロセッサ1 bの連続書き込みがほぼ同じに始まった場合、本発明の第1実施例では図4 (b) に示すようなタイミングで実行される。

【0042】図4 (b) に示すタイミングで動作するプロセッサ1 aは、先にメモリアクセスを実行するので、まだどのプロセッサも従のメモリ3への書き込みをしていないので、直列化部7を占有して、従のメモリ3への書き込みを行なう。

【0043】一方、プロセッサ1 bは、プロセッサ1 aによって直列化部7が先に占有されているので、直列化部7が解放されるまで待つ。プロセッサ1 aによる従のメモリ3への書き込みが終了すると(時間 t_1)、直列化部7が解放されてプロセッサ1 bによる従のメモリ3への書き込みが開始される。

【0044】一方、プロセッサ1 aは、従のメモリ3に対する書き込みに続いて主のメモリ4への書き込みを行っている。従って、プロセッサ1 a、1 bは、書き込みを行なうメモリが異なることから、2つのプロセッサ1 a, 1 bによる書き込みを並行して実行できるので、メモリネックによる書き込み性能の低下はない。

【0045】さらに、プロセッサ1 bによる従のメモリ3への書き込みが終了する時点(時間 t_2)では、プロセッサ1 aの主のメモリへの書き込みも終了していることから、プロセッサ1 bが主のメモリ4への書き込みを開始してもメモリネックによる書き込み性能の低下もない。

【0046】以上のことから、本発明のマルチプロセッサシステムでは、同時に連続書き込みを実行しても、従のメモリ3への直列化することにより、プロセッサ1 bによる書き込みが完了するまでの時間が単独実行時の1.5倍(時間 t_3)になるだけである。従来のメモリアクセス制御方法では、図X2に示すように、2つのプロセッサによる書き込みがそれぞれ2倍の書き込み時間を要しているのと比較すると、書き込み時間の増加を1/4に抑えられている。従って、システム全体の性能の向上が図れる。

【0047】次に、連続書き込みの直列化により、全体の書き込み時間を短縮できることについて、より具体的な状況毎に説明する。前述した例では、2つのプロセッサによる書き込みに要する時間がほぼ同じものとして説明したが、以下ではプロセッサによって処理時間がことなる場合を含めて説明する。

【0048】ここでは、2つのプロセッサによるメモリへの重なり合う2つの連続書き込みをA、Bとし、それぞれの重なりのない状態での主及び従メモリへの書き込み時間をa、b、連続書き込みAが始まってから、連続書き込みBが始まるまでの遅れ時間をdとする。以下、書き込み時間a、b、遅れ時間dの大小関係で場合分けして、直列化することで書き込み時間の増加を抑えられることを説明する。

(1) $a \leq b$ の場合。

10 【0049】(1-1) 直列化しない場合について、さらに場合分けすると、図5 (a) (b) に示すようになる。図中、斜線部分が、1つのメモリに対して2つのプロセッサによる書き込みが重なっている部分である。

【0050】図5 (a) に示す $d + b < 2b$ の場合では、書き込みA、Bは、従のメモリに対する書き込みと、主のメモリに対する書き込みの両方で重なっている。また、図5 (b) に示す $d + b \geq 2b$ の場合では、書き込みA、Bは、従のメモリに対する書き込みが重なっている。

20 【0051】図5 (a) (b) に示すように、 $a \leq b$ の場合では、少なくとも $a - d$ だけは書き込みA、Bは重なることになる。従のメモリが1つしかない場合には、このメモリがボトルネックになって、書き込みA、Bを交互に行なうことしかできない。従って、連続書き込みA、Bの両方で $a - d$ 分の待ちが発生することになり、2つの書き込みを合わせて $2(a - d)$ だけ書き込み時間が長くなる。 $d + b < 2b$ の場合では、さらに主のメモリに対する書き込みで重なった部分によって、 $2(2a - (d + b))$ 分書き込み時間が長くなる。

30 【0052】(1-2) 一方、直列化する場合は、図6に示すようになる。従のメモリに対する書き込みAと書き込みBを直列化することにより、連続書き込みBが $a - d$ だけ遅れるが、従のメモリと主のメモリの何れもアクセスが衝突しなくなる。従って、2つ合わせた書き込み時間は、 $a - d$ 分だけ長くなる。

【0053】このように、 $a \leq b$ の場合では、従メモリへの書き込みを直列化することにより、2つの書き込みA、Bを合わせた書き込み時間の増加を、少なくとも $2(a - d)$ あったものを、半分の $a - d$ に抑えることができる。

(2) $a > b$ の場合。

【0054】(2-1) 直列化しない場合について、 $a \leq 8/5b$ の範囲を場合分けして図示すると、図7

(a) (b) (c) に示すようになる。なお、 $a \leq 8/5b$ の範囲で限定するのは、一般に、連続書き込みの書き込み時間には極端なばらつきはなく、 $b \leq 5/8a$ 、つまり、bがaのほぼ半分以下になることは少なく、無視できるものとしている。

50 【0055】図7 (a) に示す $b < a \leq d/2 + b$ の場合では、書き込みA、Bは、従のメモリに対して($a -$

d) 分が重なり、主のメモリに対する書き込みが $2a - (d + b)$ 分が重なる。従って、従と主のメモリに対する書き込みの重なりは、合わせて $2(a - d) + (a - b)$ となる。図7(b)に示す $d/2 + b < a \leq d + b$ の場合では、書き込みA、Bは、従のメモリに対して $(a - d)$ 分が重なり、主のメモリに対する書き込みが b 分が重なる。従って、従と主のメモリに対する書き込みの重なりは、合わせて $(a - d) + b \geq (a - d) + (a - b)$ となる (なぜならば $a \leq 2b$)。図7(c)に示す $d + b < a \leq 8/5b$ の場合では、書き込みA、Bは、従のメモリに対しては b 分が重なり、主のメモリに対しては $d + 2b - a$ 分が重なる。従って、従と主のメモリに対する書き込みの重なりは、合わせて $3b + (d - a)$ となる。

【0056】図7(a)(b)に示すように、少なくとも $(a - d) + (a + b)$ だけは書き込みA、Bは重なることになる。従と主のメモリが1つずつしかない場合はこれらのメモリがボトルネックになって、書き込みA、Bの書き込みを交互に行なうことしかできず、連続書き込みの両方で $(a - d) + (a + b)$ 分の待ちが発生することになり、2つの書き込みを合わせて $2(a - d) + 2(a + b)$ だけ少なくとも書き込み時間が長くなる。

【0057】(2-2) 一方、直列化する場合には、図8に示すようになる。従のメモリに対する書き込みAと書き込みBを直列化することにより、連続書き込みBが $a - d$ だけ遅れ、主のメモリに対する書き込みが $a - b$ だけ重なるが、従のメモリと主のメモリの何れもアクセスが衝突しなくなるので、2つ合わせた書き込み時間は、 $a - d + 2(a - b)$ 時間だけ長くなる。

【0058】このように、 $a > b$ の場合では、従メモリへの書き込みを直列化することにより、2つの書き込みA、Bを合わせた書き込み時間の増加を、直列化しない場合では少なくとも $2(a - d) + 2(a - b)$ 分も増加していたものを、少なくとも $(a - d)$ だけに増加を抑えることができる。

【0059】また、図7(c)に示す場合に対しても、 $2(3b + d - a) - (a - d + 2(a - b)) = 8b + 3d - 5a \geq 8b - 5a \geq 0$ (なぜなら $a \leq 8/5b$) となり、従と主メモリに対する書き込み時間 a 、 b との差が37.5%以内である2つの書き込みが重なる場合には、直列化することにより書き込み時間の増加を小さく抑制することができる。

【0060】次に、本発明の第2実施例について説明する。第2実施例は、前述した第1実施例と同じ構成を持つものとするが、メモリアクセス制御方法が異なる。図6は本発明の第2実施例におけるメモリアクセス制御方法の動作を示すフローチャートである。

【0061】次に、本発明の第2実施例について説明する。第2実施例は、前述した第1実施例で説明したマル

チプロセッサシステムと同様の構成であるものとして説明を省略する。ただし、第2実施例では、直列化部7は、さらに書き込みの直列化の有効性を判定するために用いられる書き込みの開始時間と、単独で書き込みを行なったときの予想継続時間を設定できるものとする。また、連続書き込み検出部6は、さらに書き込むデータ量等から単独で行ったときの予想書き込み時間を求める機能を有しているものとする。各プロセッサ(メモリアクセス制御部10)は、直列化部7に設定された書き込みの開始時間と予想継続時間、及び連続書き込み検出部6によって求められた予想書き込み時間をもとに直列化の有効性を判定する機能が設けられる。

【0062】第1実施例におけるメモリアクセス制御方法では、後から実行される連続書き込みの継続時間がすでに実行している連続書き込みの継続時間の5/8以上であれば、常に、従のメモリに対する書き込みを直列化した方が有効であるが、後から実行する書き込みの継続時間が非常に短い場合には、直列化しない方が書き込み時間の増加が小さくなる。そこで、このような場合には直列化しないようにしたものが、第2実施例におけるメモリアクセス制御方法である。

【0063】次に、第2実施例の動作について説明する。第2実施例におけるプロセッサ1a~1dは、図9に示すフローチャートに従ってメモリアクセスを行なう。まず、メモリアクセスを行なうプロセッサ(メモリアクセス制御部10)は、連続書き込み検出部6を使って、メモリアクセスが連続書き込みか読み出しかを判断する(ステップB1)。

【0064】メモリアクセスが読み出しの場合には、プロセッサは、主のメモリ4から読み出しを行なう(ステップB2)。連続書き込みの場合には、最小メモリ構成検出部5により、主と従のメモリが1つずつしかなく、同時に連続書き込みを行なうとメモリでの転送ネックが発生する最小メモリ構成であるかを判別する(ステップB3)。

【0065】メモリが最小構成でなければ、プロセッサは(メモリアクセス制御部10による制御のもとに)、他のプロセッサが連続書き込みを行っている、いないにかかわらず、メモリに対する連続書き込み、すなわち従のメモリ3への書き込みを行ない(ステップB4)、次に主のメモリへの書き込みを行なう(ステップB5)。

【0066】一方、メモリが最小構成の場合には、従のメモリ3への連続書き込みを直列化するため、プロセッサ(メモリアクセス制御部10)は、直列化部7に問い合わせて直列化部7が空いているか、つまり他のプロセッサが従のメモリ3への連続書き込みを行なっているかを調べる(ステップB6)。

【0067】直列化部7が空いている場合には、プロセッサは、従来のメモリ3に対する書き込みを行なっている間は、他プロセッサによる従のメモリ3への連続書き

込みを禁止させるため直列化部 7 を占有する。また、プロセッサは、直列化部 7 に書き込み開始時刻と、単独で書き込みを行なったときの予想継続時間を設定する（ステップ B 7）。ここで直列化部 7 が占有されると他のプロセッサは、従のメモリ 3 に対する書き込みを実行できず、直列化部 7 が解放されるまで待ち状態となる。

【0068】続いて、プロセッサは、従のメモリ 3 への連続書き込みを実際に開始する（ステップ B 8）。プロセッサは、従のメモリ 3 に対する書き込みが完了した後で直列化部 7 を解放し（ステップ B 9）、他のプロセッサが連続書き込みを開始できるようにする。その後、主のメモリ 4 への連続書き込みを行なう（ステップ B 10）。直列化部 7 が解放されると、空くまで待っていたプロセッサが、直列化部 7 を占有して、従のメモリ 3 に対する連続書き込みを開始する。

【0069】一方、ステップ B 6 において、直列化部 7 が空いていない場合には、プロセッサ（メモリアクセス制御部 10）は、直列化部 7 を占有している他のプロセッサによる従のメモリ 3 への連続書き込みが開始された開始時刻と、その予想継続時間を直列化部 7 から読み出す（ステップ B 11）。

【0070】そして、プロセッサは、読み出した開始時刻及び予想継続時間と、これから行なう書き込みの予想継続時間から、書き込みを直列化した場合の書き込み時間と直列化しない場合の書き込み時間を計算し、その結果をもとに直列化の有効性を判定する（ステップ B 12）。

【0071】ここで有効ではないと判定された場合には、そのまま書き込みの直列化を行わずに、従のメモリに対する書き込み（ステップ B 4）、主のメモリに対する書き込み（ステップ B 5）を実行する。

【0072】一方、書き込みの直列化が有効であると判定された場合には、従のメモリに対する他のプロセッサによる書き込みが終了し、直列化部 7 が解放されるまで待つ（ステップ B 7）。直列化部 7 が解放されると、空くまで待っていたプロセッサが、直列化部 7 を占有して、従のメモリ 3 に対する連続書き込みを開始する。

【0073】ここで、第 2 実施例について、図 1 に示す最小メモリ構成のマルチプロセッサシステムで、プロセッサ 1 a による書き込み A とプロセッサ 1 b による書き込み B がほぼ同じに始まった場合について、簡単な例を用いて説明する。なお、説明の都合上、プロセッサ 1 a による連続書き込み A の方が少しだけ早く開始されたものとし、プロセッサ 1 a、1 b による書き込みに要する時間はほぼ同じものとする。

【0074】プロセッサ 1 a（あるいはプロセッサ 1 b）によって、単独で連続書き込みを実行する場合、図 10（a）に示すように、従のメモリへの書き込み、主のメモリへの書き込みを、連続して最短の時間で実行できる。

【0075】これに対し、プロセッサ 1 a とプロセッサ 1 b の連続書き込みがほぼ同じに始まった場合、従のメモリに対する書き込みを直列化すると、図 10（b）に示すようになる。すなわち、書き込み B の従のメモリに対する書き込みを遅らせても、2 つの連続書き込み A、B が最終的に完了する時間は変わらず（書き込み A の主のメモリに対する書き込みが完了した時点）、むしろ書き込み B を遅らせるための制御に時間を要してしまう。

【0076】第 2 実施例では、図 10（c）に示すように直列化しないようにして、連続書き込み B の実行を遅らせず直ちに実行する。この場合、従のメモリに対する書き込みにおいて書き込み時間の増加が発生するが、主のメモリに対する書き込みには遅れが発生しないので、結局、連続書き込み A の書き込みが完了した時点で、連続書き込み A、B の書き込みが完了する。

【0077】なお、図 9 に示すフローチャートでは、ステップ B 7 において、書き込みの開始時刻を設定しているが、この情報がなくても、それぞれの単独での予想書き込み時間から、直列化した場合と直列化しない場合の予想書き込み時間の期待値を計算し、その結果に応じて直列化が有効であるか否かを判定することもできる。この方法であっても、多くの場合は正しい判別が可能である。

【0078】このようにして、直列化を有効性を判定した上で、従のメモリに対する書き込みを直列化するので、直列化によるメモリアクセス方法を、より最適化させることが可能となる。

【0079】

【発明の効果】以上詳述したように本発明によれば、最小メモリ構成であっても、複数のプロセッサによるメモリに対する書き込みが重なった場合に、最適に書き込みの直列化を行なうので、書き込みの性能低下を小さくすることが可能となるものである。

【図面の簡単な説明】

【図 1】本発明の実施例に係わるマルチプロセッサシステムの構成を示すブロック図。

【図 2】本発明のメモリアクセス制御方法をマルチプロセッサシステム上で実現する機能構成を示すブロック図。

【図 3】本発明の第 1 実施例におけるメモリアクセス方法の動作を説明するためのフローチャート。

【図 4】第 1 実施例における連続書き込みが同時に始まった時の実行のタイミングを説明するための図。

【図 5】本発明の連続書き込みの直列化により全体の書き込み時間を短縮できることを説明するための図。

【図 6】本発明の連続書き込みの直列化により全体の書き込み時間を短縮できることを説明するための図。

【図 7】本発明の連続書き込みの直列化により全体の書き込み時間を短縮できることを説明するための図。

【図 8】本発明の連続書き込みの直列化により全体の書

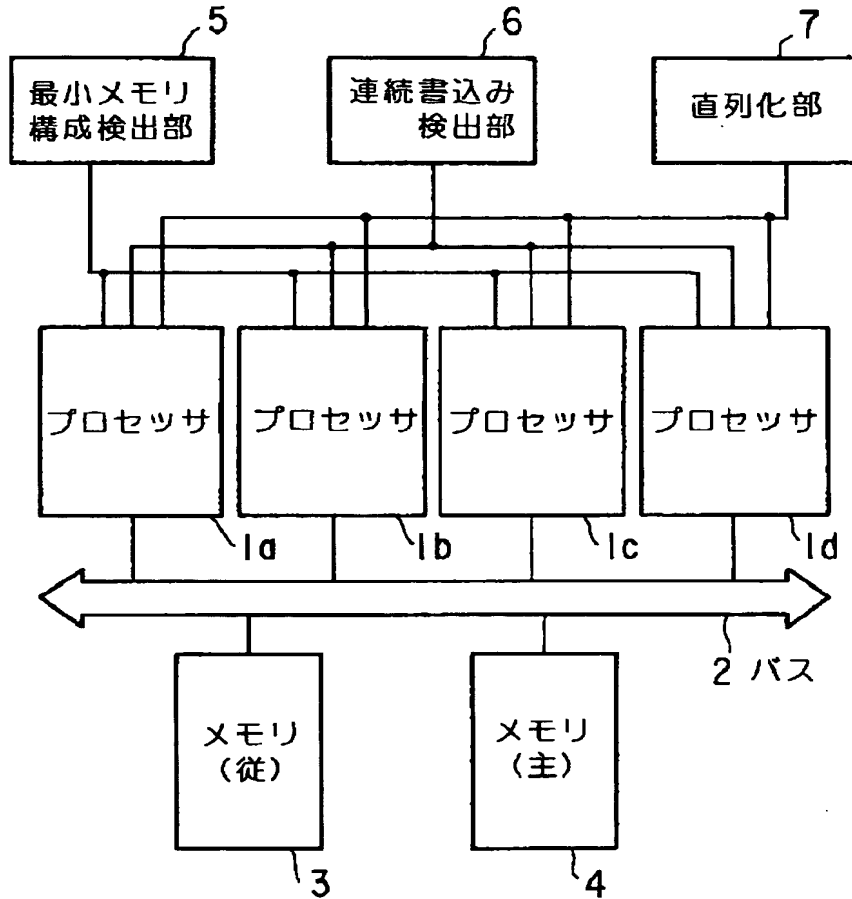
き込み時間を短縮できることを説明するための図。

【図9】本発明の第2実施例におけるメモリアクセス方法の動作を説明するためのフローチャート。

【図10】第2実施例における連続書き込みが同時に始まった時の実行のタイミングを説明するための図。

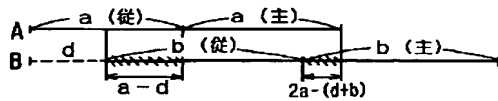
【図11】従来のメモリアクセス制御方法を説明するためのマルチプロセッサシステムの構成を示す図。

【図1】

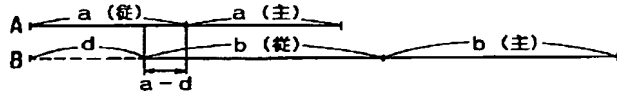


【図5】

(a) $d + b < 2b$ の場合



(b) $d + b \geq 2b$ の場合

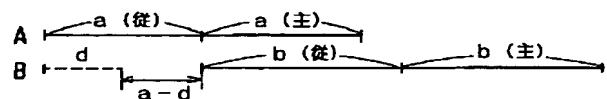


【図12】従来の連続書き込みが同時に始まった時の実行のタイミングを説明するための図。

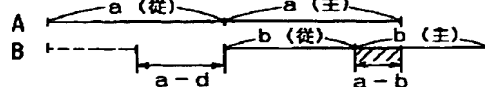
【符号の説明】

1 a ~ 1 d … プロセッサ、2 … バス、3, 4 … メモリ、5 … 最小メモリ構成検出部、6 … 連続書き込み検出部、7 … 直列化部、10 … メモリアクセス制御部、12 … ハードウェア構成情報格納部。

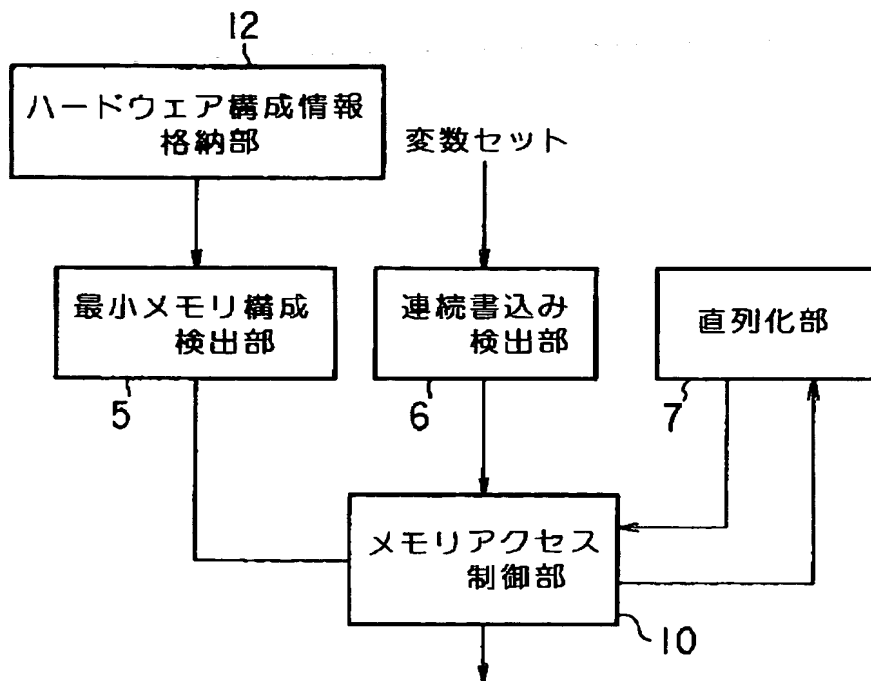
【図6】



【図8】

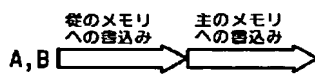


【図2】

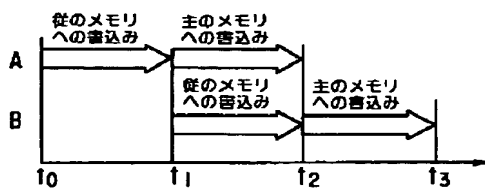


【図4】

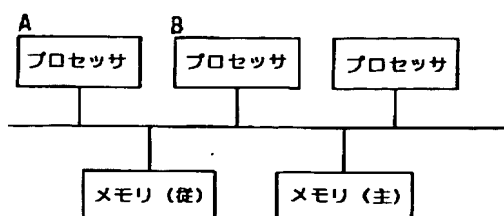
(a) 単独実行時のタイミング



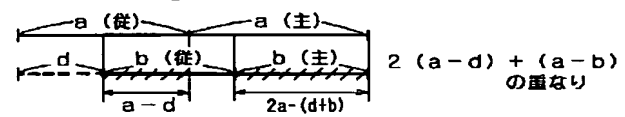
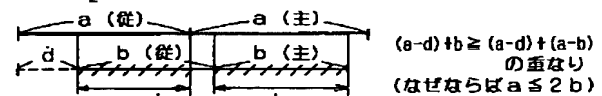
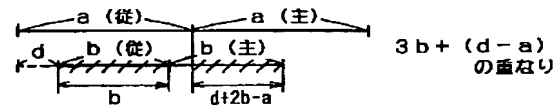
(b) 同時実行時のタイミング



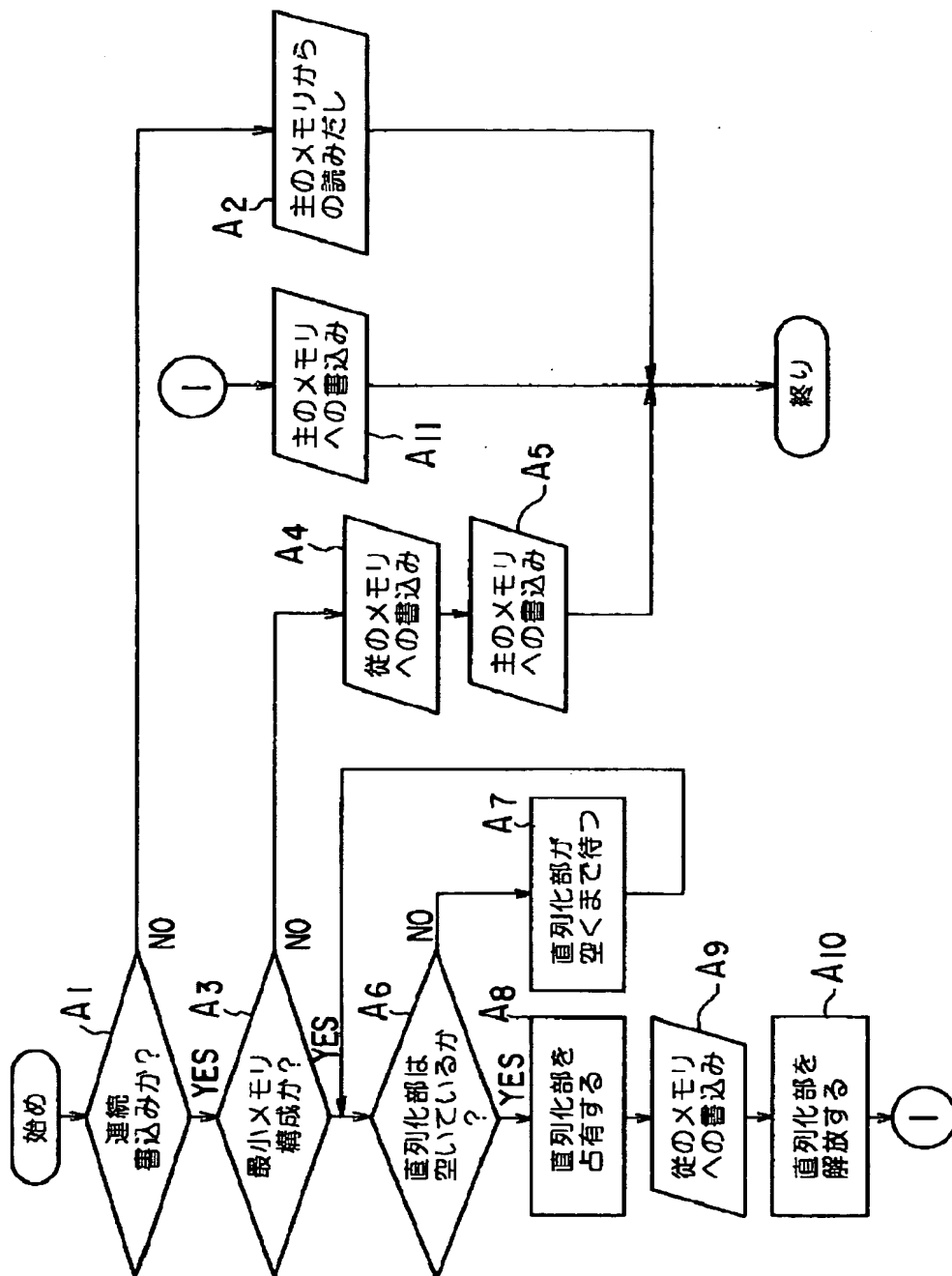
【図11】



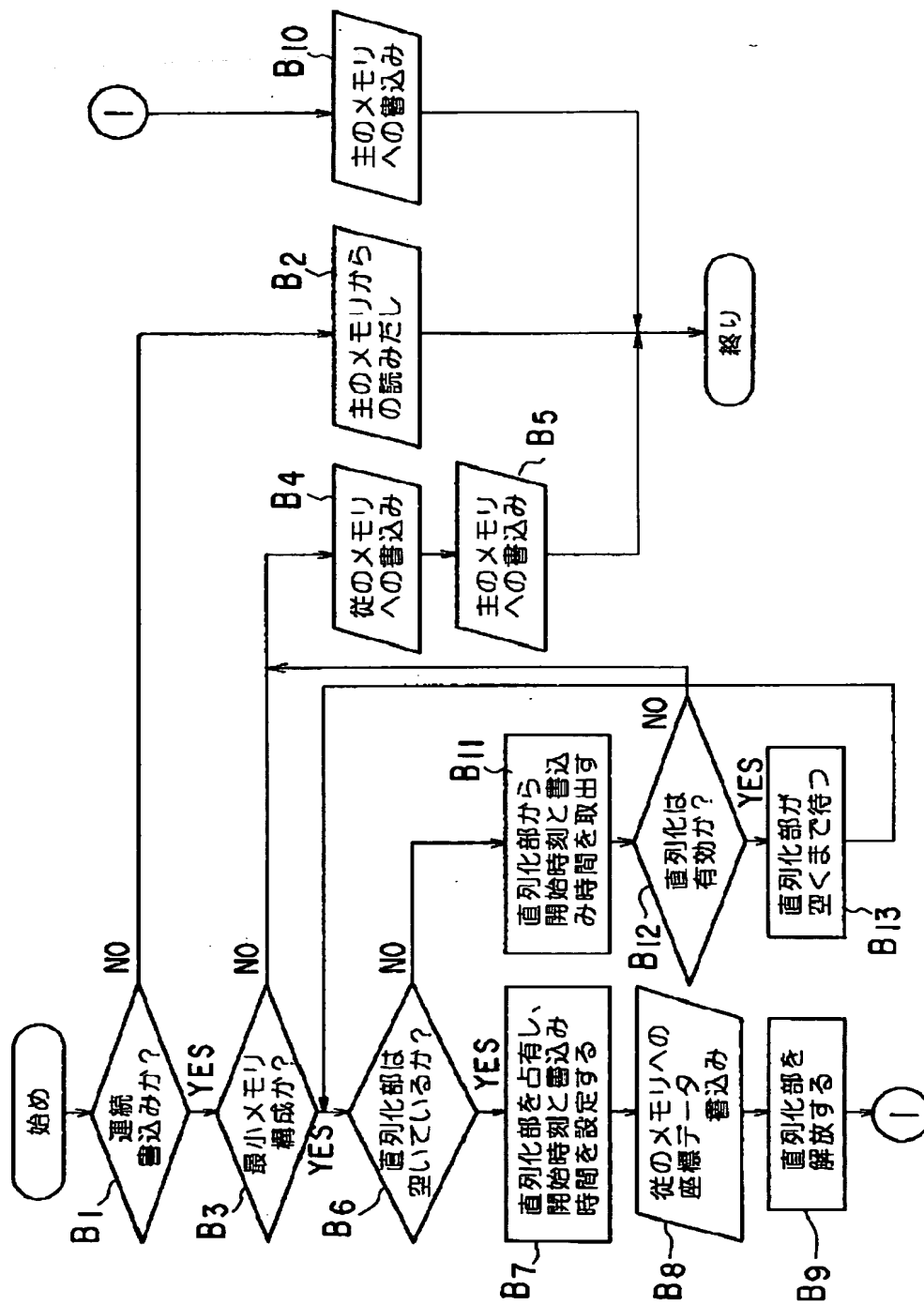
【図7】

(a) $b < a \leq \frac{d}{2} + b$ の場合(b) $\frac{d}{2} + b < a \leq d + b$ の場合(c) $d + b < a \leq \frac{8}{5}b$ の場合

【図 3】

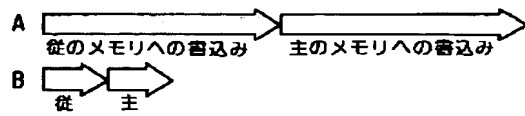


【図9】

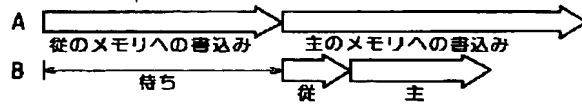


【図 10】

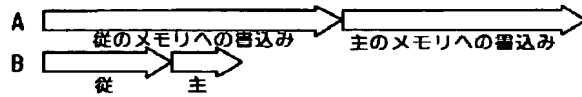
(a) 単独実行時のタイミング



(b) 同時実行時のタイミング (直列化する)

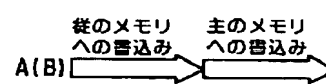


(c) 同時実行時のタイミング (直列化しない)



【図 12】

(a) 単独実行時のタイミング



(b) 同時実行時のタイミング

